This is a SUPPLEMENTAL INFORMATION SHEET to be used to provide additional information regarding the invention disclosure referenced above.

Item ___

Disclosure:
A Performance Monitoring and Management System for Real-Time Networks

David Grabelsky
Ikhlaq Sidhu
Guido Schuster
Jacek Grabiec

Advanced Technologies Department
Carrier Systems Business Unit, 3Com

Abstract

Gateway routers for real-time networks have the ability to collect delay, loss, and jitter statistics on a per-connection basis. It is possible to use this information not only to monitor the quality of individual voice calls and other real-time connections, but to evaluate the overall performance of the underlying network. This paper presents describes a method for monitoring and managing a real-time data network that supports voice and other real-time services. It is proposed: 1) that the RTCP mechanisms of RTP for sender and receiver reporting be used to relay performance information to one or more network monitoring sites for analysis and interpretation; 2) that a gateway routers are organized and managed within a hierarchy that allows the monitoring function to localize it view of network conditions within the hierarchy; and 3) that monitoring occurs on various time scales.

Introduction

Network edge devices that perform processing for real-time services can fairly easily collect statistics on delay, packet loss, and jitter on a per-connection basis. In fact, these statistics are collected routinely as part of dynamic buffering schemes [1]. On an individual connection, these data provide a means of monitoring the quality of service being provided. On a global scale, the statistics from all connections can be used to monitor the overall performance of the underlying real-time network, providing a picture of "average" network conditions, as well as highlighting trouble spots. It is therefore desirable to develop a network performance monitoring and management system based upon the per-connection statistics collected by the edge devices.

This disclosure describes such a system. The edge device is a gateway router. In the context of Voice over IP (VoIP), these devices are referred to as the voice gateways, and the data are transported in RTP streams. The RTP specification [2] includes a control protocol called Real Time Control Protocol, or RTCP. A large part of RTCP is aimed at collecting statistics on the quality of the transport service between session members; i.e., remote applications communicating via RTP streams. The design presented here utilizes RTCP to collect/measure the relevant statistics, and defines how they are used to build the databases which will support the network management system. For purposes of illustration, only a voice network is considered here. However, the basic design applies to any real-time network; i.e., one which carries video, supports conference calling, multicasting, etc.

inventors' initials ______ ______ ______ ______ ______

High-Level Architecture

Figure 1 shows a simplified picture of the high-level architecture of the VoIP system and network. Analog phone calls are terminated at modems in edge devices called voice gateways. For each call, incoming analog (voice) data are sampled, coded, and packetized by a dedicated modem in one voice gateway, then forwarded by a router in the gateway onto the IP network to another, remote voice gateway. At the remote gateway the packets are routed from the IP network to specific modems, according to an appropriate ID for the phone call. The data in each packet are decoded and played out, recreating the original analog signal (with some associated fidelity), and finally transmitted to the receiving telephone. (For simple two-way phone calls, this process is obviously symmetric.) Omitted from this figure are the architectural components needed for signaling, network admission control, etc.

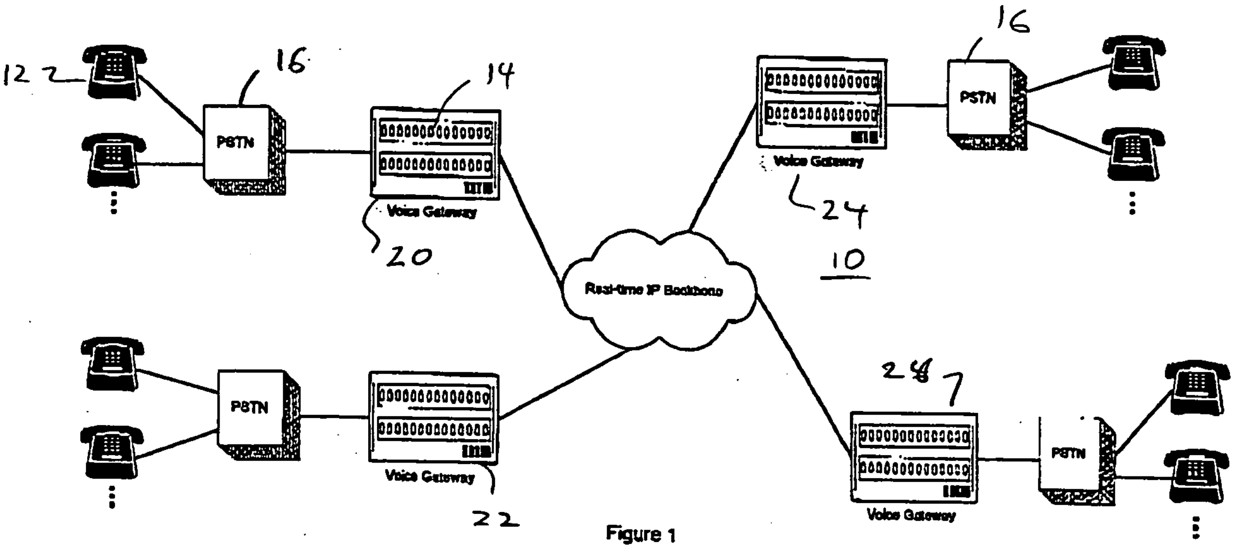

Figure 1

The coded data are packetized in RTP packets, which are themselves transported in UDP packets on the IP network. RTP is designed to optimize the end-system processing for the real-time nature of the data carried in the RTP packets. A stream of RTP packets which is associated with a given phone call is said to belong to an RTP session. That is, the session identifies the call, and session members participate in the call. (The concept of RTP session is broader than just phone call ID. For example, multiple media streams associated with a single video conference would comprise multiple RTP sessions, with multiple participants per session. For the purposes of the current document, the scope of an RTP session is limited to simple two-way phone calls.) For a given session, each participant is classified within the RTP protocol as either a sender or receiver, based upon how recently it has transmitted an RTP packet.

In addition to support for real-time data, RTP includes a control protocol, RTCP, which allows session members to exchange information related to performance, as well as to various signaling functions. RTCP data are carried in RTCP packets. These are distinct from RTP packets, but are transported on the same lower-layer protocol (UDP on IP). Each RTP session includes some proportion of RTCP traffic. That is, RTCP packets associated with an RTP session are transmitted "in-band." Of particular relevance to the design of a network management system are RTCP sender reports (SRs) and receiver reports (RRs). SRs and RRs carry information that can be used to characterize conditions on the IP network carrying the RTP traffic. Depending upon whether a session member is a sender or a receiver, it periodically transmits an SR or an RR to all other session members (only one, in the simple VoIP model considered here). Thus, each session member periodically receives an SR or RR from the other session member(s).

Every SR and RR includes one reception block for every source from which the creator of the SR or RR is receiving RTP packets. For a simple two-way voice call, each session member receives from just one source; i.e., the other end of the call. An SR includes, in addition to the reception block(s), a sender information block. The data in a reception block describe statistical properties of RTP packet reception as observed by the creator of the SR or RR. The sender information block includes data that describe statistical properties of packet transmission as reported by the creator of the SR. Thus when endpoint A receives an RR from endpoint B, A can determine how well B "hears" A. For example, A can compare its received packet count with the number of packets expected (as inferred from RTP sequence numbers) and thus determine the fractional packet loss. When endpoint A receives an SR from endpoint B, A gets the same information as in an RR, and in addition gets information on how many RTP packets B has sent to A. These data can be used to compute throughput at B. Note that a reception block contains most of the information relevant to the quality of the connection.

The relation between the high-level picture shown in Figure 1 and RTP sessions and streams is shown in Figure 2. Each pair of lines between two gateways represents an RTP session terminated at the gateways. As illustrated in this figure, a one gateway may simultaneously terminate sessions with several other gateways. It is expected that the network conditions will, on average, be the same for all sessions between a given pair of gateways. This suggests that copies of the reception blocks from all SRs and RRs created at a given gateway for transmission be maintained at the gateway. Further, these reception blocks should be partitioned according to source gateways to which they apply, and be processed as a statistical ensemble to provide a characterization of the network conditions between that source-destination gateway pair. The next section discusses two alternative approaches to retention of reception blocks by the gateway that generates them.

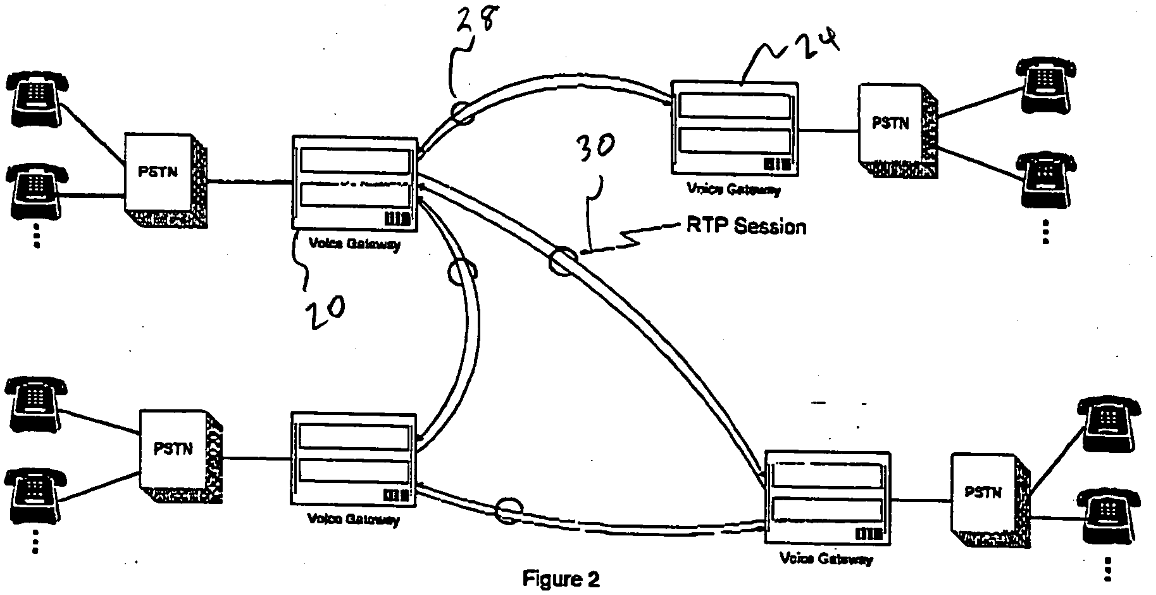

Figure 2

In order to provide a network-wide view of conditions, the monitoring system will include a number of processing and monitoring sites for collecting the data on all three time scales. These sites will be organized in a topological hierarchy that is defined according to groups of gateways. The hierarchy uses a naming convention of *clusters* to refer to groupings of gateways, and *levels* to define placement within the hierarchy. At the lowest level of the hierarchy we define a unit called a *Level_0 cluster* comprised of a set of gateways, called *Level_0 cluster members*. Each gateway in a Level_0 cluster may talk to any other gateway in the Level_0 cluster, so that the Level_0 cluster also defines every possible gateway pair that can be formed by its members. The term *Level_0 cluster pair* is used to define a gateway pair for two gateways belonging to the *same* Level_0 cluster. (The case of multiple, co-located gateways is considered a single, compound gateway in this model.) The monitor for given Level_0 cluster is responsible for monitoring the network conditions between all Level_0 cluster pairs in its Level_0 cluster. The term *Level_0 cluster monitor* is used to define this monitor.

The next higher level of the hierarchy defines a unit called a *Level_1 cluster*. Each Level_1 cluster consists of a specific set of Level_0 clusters, called *Level_1 cluster members*. We define a *Level_1 cluster pair* as a pair of gateways formed by any two gateways belonging to *different* Level_0 clusters within the Level_1 cluster. That is, each gateway in a Level_1 cluster pair belongs to a different Level_0 cluster, but each gateway's parent Level_0 cluster is a member of the Level_1 cluster. A Level_1 cluster has a distinct monitor for all Level_1 cluster pairs within the Level_1 cluster. The term *Level_1 cluster monitor* is used to define this monitor.

The next higher level of the hierarchy defines a unit called a *Level_2 cluster*. Each Level_2 cluster consists of a specific set of Level_1 clusters, called *Level_2 cluster members*. We define a *Level_2 cluster pair* as a pair of gateways formed by any two gateways belonging to *different* Level_1 clusters within the Level_2 cluster. That is, each gateway in a Level_2 cluster pair belongs to a different Level_1 cluster, but each gateway's parent Level_1 cluster is a member of the Level_2 cluster. A Level_2 cluster has a distinct monitor for all Level_2 cluster pairs within the Level_2 cluster. The term *Level_2 cluster monitor* is used to define this monitor.

The hierarchy continues up in this fashion, conceptually forever, to *Level_n*. Figure 3 illustrates the hierarchy up to the Level_2 cluster level. Note that regardless of the level of a given gateway pair, the statistics on the sessions terminated at the gateways that form the pair are collected at the each gateway. That is, the RTCP SRs and RRs are exchanged between the two gateways that form the pair. Also, no inherent restriction is placed on ability of any gateway to talk to any other gateway based upon this hierarchy. (Restrictions could be imposed if this hierarchy is also used as a basis for a distributed gatekeeper.)

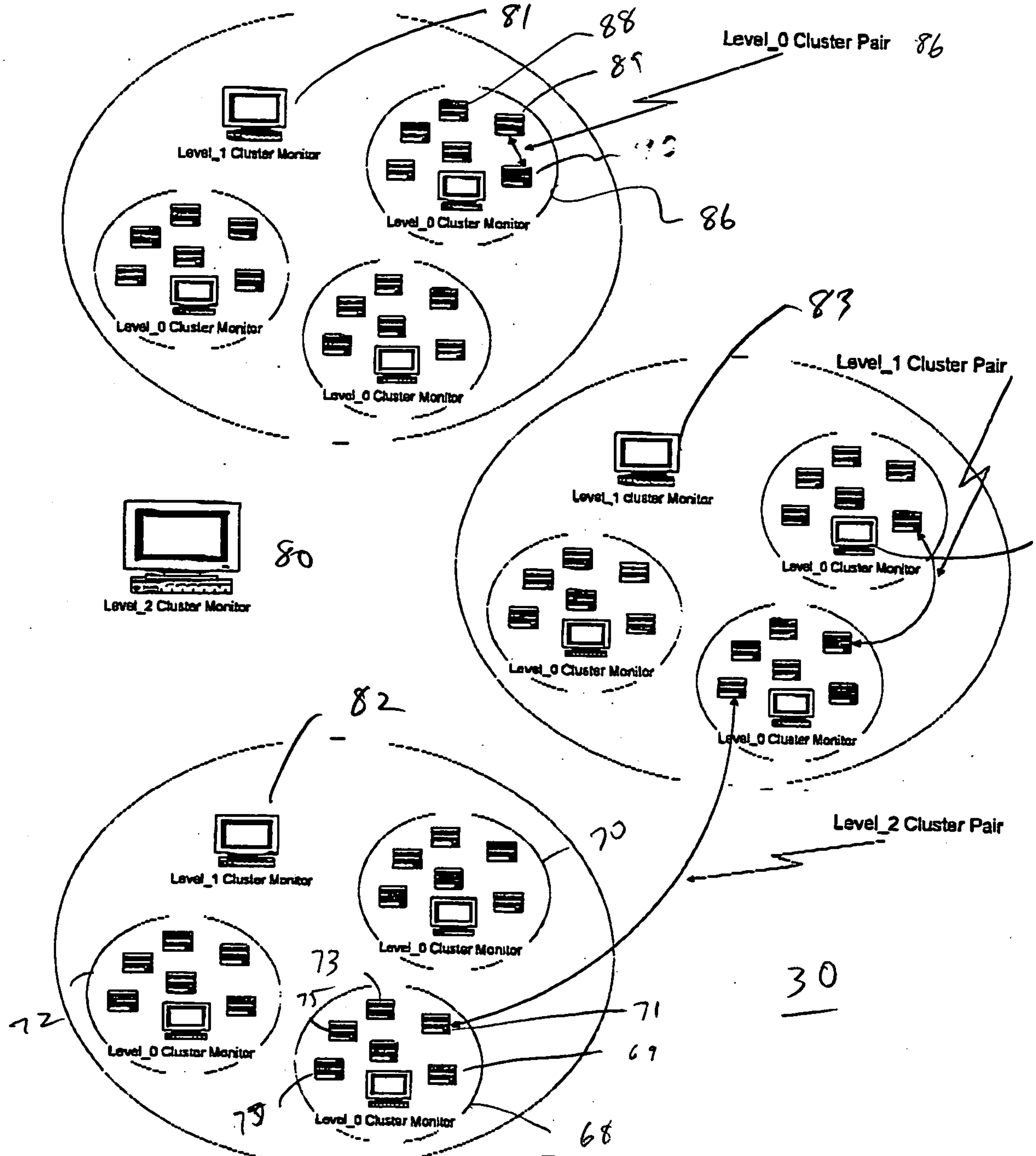

Figure 3

The purpose of this hierarchy is twofold. First, it distributes the traffic associated with the monitoring function, as well as the processing burden placed on the monitors themselves. Second, it allows for topological localization of the network conditions being monitored. In particular, problems can be traced to the smallest relevant level of the hierarchy, helping to isolate trouble spots. For a system-wide monitoring, local conditions, e.g., at the Level_0 cluster level, must be made available at the gl bal level. Thus, information must be passed up the hierarchy at some point. Related to this is the time scale on which data are collected and analyzed. Data collection time scale and processing are described in the next section.

Data Collection and Processing

Standard RR and SR Features

The interval between successive RRs and/or SRs from a given session member is adjusted dynamically in order to maintain an upper bound of 5% on the fraction of the session bandwidth consumed for RTCP traffic. As described in [2], the adjustment algorithm attempts to scale the interval linearly with the number of session members such that the fraction of session bandwidth dedicated to RTCP traffic is kept to this 5% constant. (The algorithm also includes safeguards against RTCP packet flooding which can be precipitated by sudden changes in the number of session members, in the case, e.g., of a conference call.) The actual interval used is the maximum of the computed upper limit and 5 seconds. For a two-way voice call, a very rough estimate of RTCP packet transmission interval can be obtained as follows. We assume that two 30 ms samples are transmitted in one RTP packet every 60 ms. For a lower limit on the interval (upper limit on frequency) we assume 100% utilization, and apply the 5% limit. This yields one RTCP packet every 1.2 seconds, clearly below the 5-second limit, so the 5-second interval would be applied. For utilizations of 10% and 5%, the limit on RTCP traffic yields an RTCP packet every 12 and 24 seconds, above the 5-second limit. Therefore we expect the transmission interval for RTCP packets to have an approximate range of 5 to 30 seconds, for utilizations between 100% (extreme case) and 5%. This sets the timing resolution for monitoring a single voice call carried via RTP. For a voice gateway terminating several calls from another gateway, the ensemble of calls can provide even higher time resolution for the network path between the two gateways, assuming the RTCP transmission times for all the sessions are uncorrelated.

The monitoring system will provide a view of network conditions and performance on three time scales: real-time, for alarm conditions; near real-time, current conditions; and daily, for long-term trend analysis. Alarm conditions will be sent to the monitor as soon as possible after they are discovered. From the preceding paragraph, this corresponds to a minimum time scale on the order of about 10 seconds, assuming the problem is discovered during a single call. This time can be made shorter if the same problem is encountered by several concurrent calls between two gateways. The time scale for near real-time monitoring should be set such that the fastest, "non-pathological" trends may be temporally resolved. The long-term trend analysis based upon daily monitoring data may be used for capacity planning , or other network adjustments.

As noted in the previous section, most of the information relevant to the monitoring system is available in the reception blocks of SRs or RRs. Most of the information is equally useful both to the system that generates the reception block, as well as the system that receives the reception block. That is, given two systems, A and B, then reception blocks generated by A describe how well A "hears" B, and vice versa. Thus any system can look at a reception block and extract the useful information. However, there are certain quantities that can only be computed by the system to which the reception block applies. I.e., when system B gets the reception block from system A, there is some specific information that only system B can use. An example is round trip delay. In this case, system B puts a time stamp on its SRs. System A maintains the delta time between reception of B's SR and A's transmission of its next reception block back to B. Both the time stamp and delta time included in A's reception block to B. When B gets the reception block from A, B subtracts these included times from its own reception time of the SR or RR that contained the block in order to determine the round trip delay. The point here is that the time stamp and delta time are only useful to system B, since these values must be compared with B's clock. Other quantities, such as packet loss and jitter observed at A are useful to any system with access to the reception block. In the following descriptions of reception block processing, it is assumed that the system which generates the reception block also computes round trip time, and that this information is made available along with the reception block to the monitor function.

RR and SR Processing Specific to this Disclosure

A copy of every reception block generated at a given gateway for inclusion in an SR or an RR should be retained by the gateway for its monitoring functions. This can be accomplished in two ways. As part f the process of generating the reception block, a copy could be "diverted" to a monitoring process. Alternatively, the full SR or RR could be generated and transmitted, with the originating gateway being designated as an RTCP "third party monitor." This configuration is described in the RTP draft [2], and requires implementation of multicasting. The advantage of the first method is that is simple and allows accumulation of reception blocks independently of full SR or RR overhead at the monitor; also, it does not require multicasting. Also, this approach could allow alarm conditions to be detected as part of the process of reception block generation, even before the full SR or RR is created. The advantage of the second method is that it standardizes the collection of reception blocks within the context of the RTCP third party monitor configuration. This could be important if interoperability becomes an issue. These two methods are illustrated in the next two figures.

A flowchart describing the high-level processing for the first method is shown in Figure 4. When an RR or SR associated with a session on a gateway is generated, a copy of the included reception block(s) are made available to Phase 0 processing. This tests delay, packet loss, and jitter against alarm thresholds. An alarm condition cause the process to trap, and an external alarm processing routine to be invoked. This eventually results in a message being sent to the monitor for the associated gateway pair. After execution resumes, Phase 1 processing does the near real-time processing. This updates the statistics for the source gateway to which the reception block applies. Finally, Phase 2 processing does the long-term processing. The specific steps included in the each phase of the processing, and how they are implemented is still under study. Note that the gateway on which this process executes is the same one on which the SR or RR is generated. That is, even though an SR or RR is intended for transmission to an external gateway, the information in the reception block refers to network conditions as observed at the originating gateway. Therefore, the originating gateway keeps a copy of the reception block and reports results to the appropriate monitor.

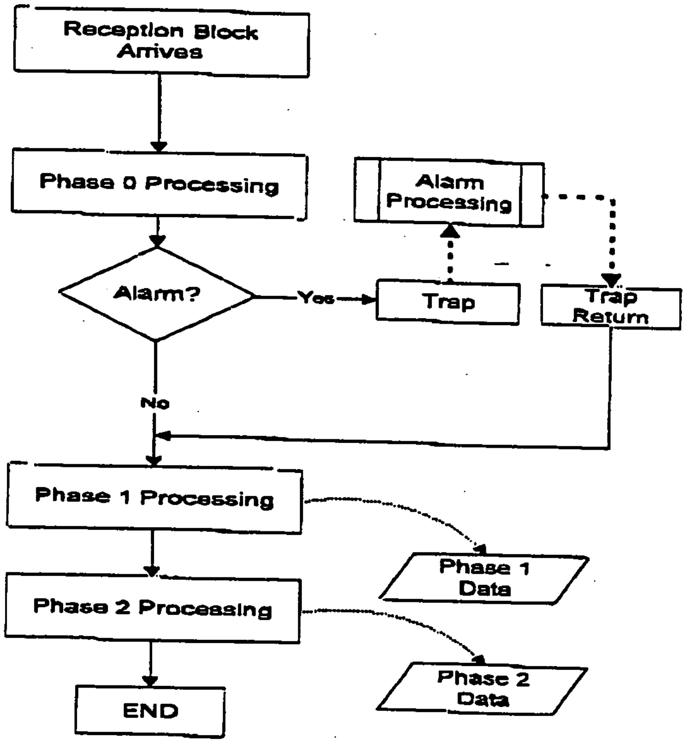

Figure 4

A flowchart describing the high-level processing for the second method is shown in Figure 5. In this approach, the entire RTCP packet is generated and, in addition to being transmitted externally, is sent to an internal process on the originating gateway. If the packet is determined to be an SR or RR, its reception block is extracted, then processing proceeds as in the first method (Figure 4). This process could also execute on an external, third-party monitor, as described in the RTP draft document [2]. Such a method might more easily accommodate interoperability of different vendors' monitoring equipment. The inclusion of the full RTCP packet in this figure also allows for the possibility that other monitoring functions may added later using other RTCP packet types. As with Figure 4 above, it is assumed (at least for now) that it is the originating gateway that processes the RTCP packet that an earlier process created for external transmission.

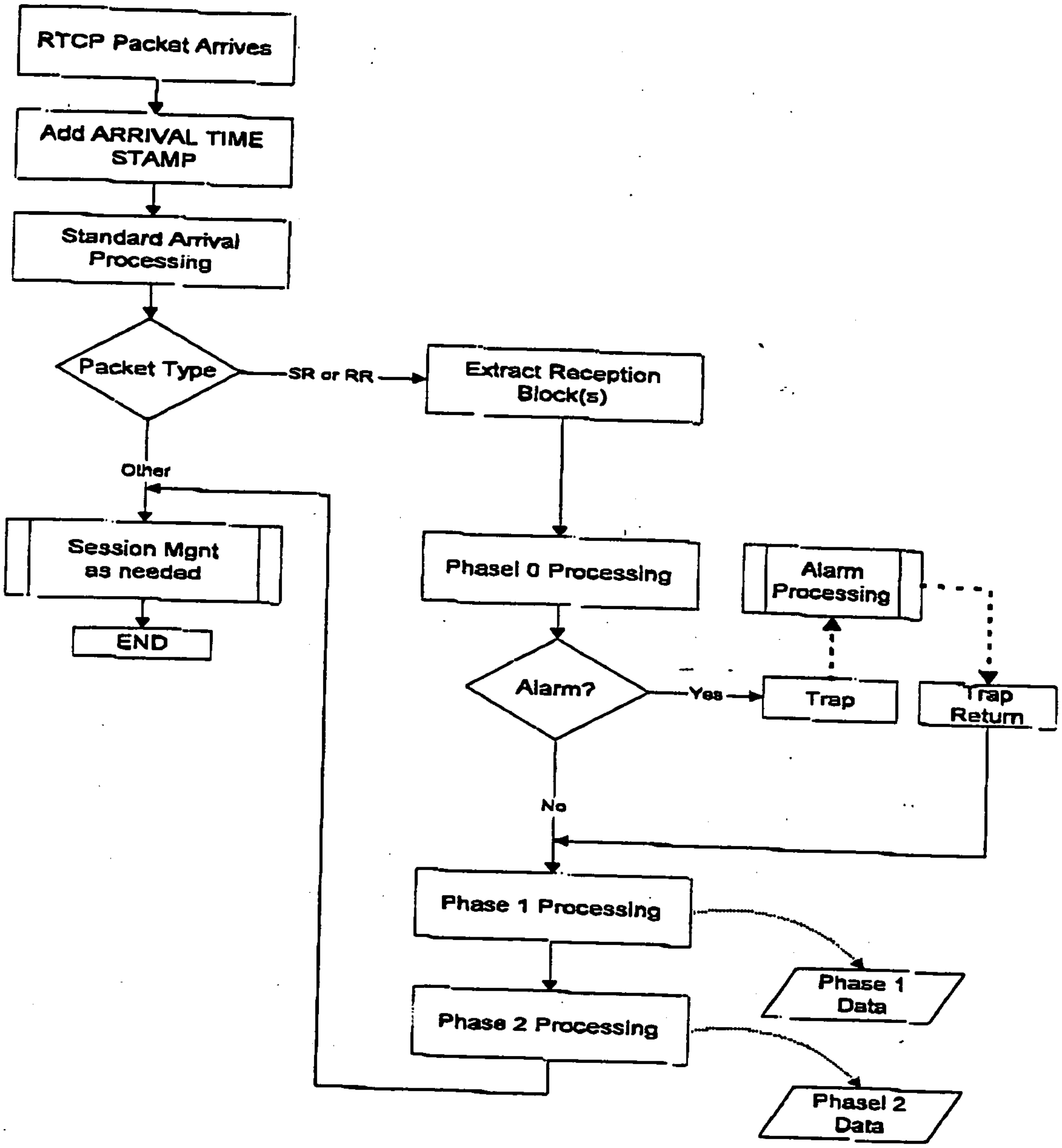

Figure 5

The next three subsections provide some more detail about the Level 0, 1 , and 2 processing.

Phase 0 Proc ssing

Phase 0 processing determines the following quantities on a per session basis:

- Round trip delay

- Jitter
- Packet loss: fraction and cumulative
- Receive buffer length
- Current coder (if hypothetical coding is implemented)

If any of the first four of these exceed a specified threshold, then execution traps and to an external alarm processing routine is invoked. This routine is a separate process which logs the event and causes a message to be sent to the monitor. A yet-to-be-specified algorithm will be used to avoid flooding the network and the monitor with alarm messages. Once the alarm routine completes, execution of the RTCP receive processing resumes.

The first three items are obtained from the reception block which was generated as part of the RTCP processing RTP. Receive buffer length is assumed to be a known system parameter (not necessarily the same value for each session).

Note that the alarm processing may be divided into more than one process, so that execution after the trap may resume before the actual alarm message is sent. E.g., event logging could be combined with the queuing of a message generation and transmission routine which runs at some later time; this second routine would include the algorithm to avoid flooding.

Phase 1 Processing

Phase 1 processing uses the Phase 0 data to maintain and update a Phase 1 data base. This data base organizes the data according to gateway pair. That is, according to the source gateway in the reception block. The combination of the source gateway and the gateway on which the data base is maintained defines the gateway pair, and thus the associated monitor (cluster, supercluster, etc.). The updating process includes adding the new statistics to any previous statistics for this gateway pair. For the first four quantities, this is just an accumulation process. The appropriate way to record the current decoder (if applicable) is to be determined.

Periodically (period to be determined), the data associated with each gateway pair is transmitted to the monitor associated with the pair. A suggested period is three minutes, the average length of a voice call. The statistics are accumulated for the first four items such that at the end of each period, each updated quantity represents a time average over the period. After transmission, the statistics are reset in preparation for accumulation during the next period.

Phas 2 Processing

Phase 2 processing uses the Phase 0 data to maintain and update a Phase 2 data base. This data base is an accumulation of the raw data from each session over a long interval. The Phase 2 data base may also include the statistics from each period of Phase 1 processing. The interval of the Phase 2 data base is suggested to be on the order of one day. At the end of the Phase 2 period, the data base is transferred to the highest level monitor for the system. At this site, the monitor can track long-term trends on a system-wide basis. Transfer of the (daily) Phase 2 data may be by FTP during some relatively quiet time on the network.

Data Formats

RTCP specifies the following formats for round-trip delay, jitter, and packet loss:

- RTT: this is computed using 32-bit time values
- Jitter: 32 bits
- Packet loss: 8 bits for fractional loss; 24 bits for cumulative

Buffer length can be specified in 24 bits (or less). An 8-bit code for coder is sufficient. The combination of buffer length and coder is then 32 bits. The total length data is then 16 bytes.

Additional information includes the IP address of each end of the gateway pair. Since one of the gateways in the pair is the source of any transmission to a m nitor, its IP address is already contained in the IP header of transmission. Thus only one additional 32-bit value is needed to include the IP address of the gateway at the other end.

inventors' initials ______ ______ ______ ______ ______

The above information can be represented in a 20-byte block.